Efficient and Consistent Adversarial Bipartite Matching

Rizal Fathony [*1] **Sima Behpour** [*1] **Xinhua Zhang** [1] **Brian D. Ziebart** [1]

Abstract

Many important structured prediction problems, including learning to rank items, correspondence-based natural language processing, and multi-object tracking, can be formulated as weighted bipartite matching optimizations. Existing structured prediction approaches have significant drawbacks when applied under the constraints of perfect bipartite matchings. Exponential family probabilistic models, such as the conditional random field (CRF), provide statistical consistency guarantees, but suffer computationally from the need to compute the normalization term of its distribution over matchings, which is a #P-hard matrix permanent computation. In contrast, the structured support vector machine (SSVM) provides computational efficiency, but lacks Fisher consistency, meaning that there are distributions of data for which it cannot learn the optimal matching even under ideal learning conditions (i.e., given the true distribution and selecting from all measurable potential functions). We propose adversarial bipartite matching to avoid both of these limitations. We develop this approach algorithmically, establish its computational efficiency and Fisher consistency properties, and apply it to matching problems that demonstrate its empirical benefits.

1 Introduction

How can the elements from two sets be paired one-to-one to have the largest sum of pairwise utilities? This *maximum weighted perfect bipartite matching* problem is a classical combinatorial optimization problem in computer science. It can be formulated and efficiently solved in polynomial time as a linear program or using more specialized *Hungarian algorithm* techniques (Kuhn, 1955). This has made it an attractive formalism for posing a wide range of problems, including recognizing correspondences in similar images (Belongie et al., 2002; Liu et al., 2008; Zhu et al., 2008; Rui et al., 2007), finding word alignments in text (Chan & Ng, 2008), and providing ranked lists of items for information retrieval tasks (Amini et al., 2008).

Machine learning methods seek to estimate the pairwise utilities of bipartite graphs so that the maximum weighted complete matching is most compatible with the (distribution of) ground truth matchings of training data. When these utilities are learned abstractly, they can be employed to make predictive matchings for test samples. Unfortunately, important measures of incompatibility (e.g., the Hamming loss) are often non-continuous with many local optima in the predictors' parameter spaces, making direct minimization intractable. Given this difficulty, two natural desiderata for any predictor are:

- **Efficiency:** learning from training data and making predictions must be computed efficiently in (low-degree) polynomial time; and
- **Consistency:** the predictor's training objectives must also minimize the underlying Hamming loss, at least under ideal learning conditions (given the true distribution and fully expressive model parameters).

Existing methods for learning bipartite matchings fail in one or the other of these desiderata; exponentiated potential fields models (Lafferty et al., 2001; Petterson et al., 2009) are intractable for large sets of items, while maximum margin methods based on the hinge loss surrogate (Taskar et al., 2005a; Tsochantaridis et al., 2005) lack Fisher consistency (Tewari & Bartlett, 2007; Liu, 2007). We discuss these limitations formally in Section 2.

Given the deficiencies of the existing methods, we contribute the first approach for learning bipartite matchings that is both computationally efficient and Fisher consistent. Our approach is based on an adversarial formulation for learning (Topsøe, 1979; Grünwald & Dawid, 2004; Asif et al., 2015) that poses prediction-making as a data-constrained zero-sum game between a player seeking to minimize the expected loss and an adversarial data approximator seeking to maximize the expected loss. We present two approaches for solving the corresponding zero-sum game arising from our formulation: (1) using the double

[*] Equal contribution [1] Department of Computer Science, University of Illinois at Chicago. Correspondence to: Rizal Fathony <rfatho2@uic.edu>, Sima Behpour <sbehpo2@uic.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

oracle method of constraint generation to find a sparsely-supported equilibrium for the zero-sum game; and (2) decomposing the game's solution into marginal probabilities and optimizes these marginal probabilities directly to obtain an equilibrium saddle point for the game. We then establish the computational efficiency and consistency of this approach and demonstrate its benefits experimentally.

2 Previous Inefficiency and Inconsistency

2.1 Bipartite Matching Task

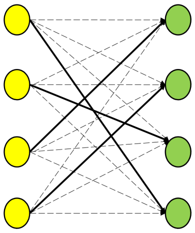

Figure 1. $n = 4$ bipartite matching task.

Given two sets of elements A and B of equal size ($|A| = |B|$), a maximum weighted bipartite matching π is the one-to-one mapping (e.g., Figure 1) from each element in A to each element in B that maximizes the sum of potentials: $\max_{\pi\in\Pi} \psi(\pi) = \max_{\pi\in\Pi} \sum_i \psi_i(\pi_i)$. Here $\pi_i \in [n] := \{1, 2, \ldots, n\}$ is the entry in B that is matched with the i-th entry of A. The set of possible solutions Π is simply all permutation of $[n]$. Many machine learning tasks pose prediction as the solution to this problem, including: word alignment for natural language processing tasks (Taskar et al., 2005b; Padó & Lapata, 2006; MacCartney et al., 2008); learning correspondences between images in computer vision applications (Belongie et al., 2002; Dellaert et al., 2003); protein structure analysis in computational biology (Taylor, 2002; Wang et al., 2004); and learning to rank a set of items for information retrieval tasks (Dwork et al., 2001; Le & Smola, 2007). Thus, learning appropriate weights $\psi_i(\cdot)$ for bipartite graph matchings is a key problem for many application areas.

2.2 Performance Evaluation and Fisher Consistency

Given a predicted permutation, π', and the "ground truth" permutation, π, the **Hamming loss** counts the number of mistaken pairings: $\text{loss}_{\text{Ham}}(\pi, \pi') = \sum_{i=1}^{n} 1(\pi'_i \neq \pi_i)$, where $1(\cdot) = 1$ if $\cdot$ is true and 0 otherwise. When the "ground truth" is a distribution over permutations, $P(\pi)$, rather than a single permutation, the (set of) **Bayes optimal** prediction(s) is: $\text{argmin}_{\pi'} \sum_\pi P(\pi)\ \text{loss}_{\text{Ham}}(\pi, \pi')$. For a predictor to be **Fisher consistent**, it must provide a Bayes optimal prediction for any possible distribution $P(\pi)$ when trained from that exact distribution using the predictor's most general possible parameterization (e.g., all measurable functions ψ for potential-based models).

2.3 Exponential Family Random Field Approach

A probabilistic approach to learning bipartite graphs uses an exponential family distribution over permutations, $P_\psi(\pi) = e^{\sum_{i=1}^n \psi_i(\pi_i)}/Z_\psi$, trained by maximizing training data likelihood. This provides certain statistical consistency guarantees for its marginal probability estimates (Petterson et al., 2009). Specifically, if the potentials ψ are chosen from the space of all measurable functions to maximize the likelihood of the true distribution of permutations $P(\pi)$, then $P_\psi(\pi)$ will match the marginal probabilities of the true distribution: $\forall i, j, P_\psi(\pi_i = j) = P(\pi_i = j)$. This implies Fisher consistency because the MAP estimate under this distribution, which can be obtained as a maximum weighted bipartite matching, is Bayes optimal.

The key challenge with this approach is its computational complexity. The normalization term, Z_ψ, is the permanent of a matrix defined in terms of exponentiated potential terms: $Z_\psi = \sum_\pi \prod_{i=1}^n e^{\psi_i(\pi_i)} = \text{perm}(\mathbf{M})$ where $M_{i,j} = e^{\psi_i(j)}$. For sets of small size (e.g., $n = 5$), enumerating the permutations is tractable and learning using the exponential random field model incurs a run-time cost that is acceptable in practice (Petterson et al., 2009). However, the matrix permanent computation is a *#P-hard* problem to compute exactly (Valiant, 1979). Monte Carlo sampling approaches are used instead of permutation enumeration to maximize the data likelihood (Petterson et al., 2009; Volkovs & Zemel, 2012). Though exact samples can be generated efficiently in polynomial time (Huber & Law, 2008), the number of samples needed for reliable likelihood or gradient estimates makes this approach infeasible for applications with even modestly-sized sets of $n = 20$ elements (Petterson et al., 2009).

2.4 Maximum Margin Approach

Maximum margin methods for structured prediction seek potentials ψ that minimize the training sample hinge loss:

$$\min_\psi \mathbb{E}_{\pi\sim\tilde{P}} \left[\max_{\pi'} \{\text{loss}(\pi, \pi') + \psi(\pi')\} - \psi(\pi) \right], \quad (1)$$

where $\tilde{P}$ is the empirical distribution. Finding the optimal ψ is a convex optimization problem (Boyd & Vandenberghe, 2004) that can generally be tractably solved using constraint generation methods as long as the maximizing assignments can be found efficiently. In the case of permutation learning, finding the permutation π' with highest hinge loss reduces to a maximum weighted bipartite matching problem and can therefore be solved efficiently.

Though computationally efficient, maximum margin approaches for learning to make perfect bipartite matches lack **Fisher consistency**, which requires the prediction $\pi^* = \text{argmax}_\pi\ \psi(\pi)$ resulting from Equation (1) to minimize the expected risk, $\mathbb{E}_{\pi\sim\tilde{P}} [\text{loss}(\pi, \pi')]$, for all distributions $\tilde{P}$. We consider a distribution over permutations that is an extension of a counterexample for multiclass classification consistency analysis with no majority label (Liu, 2007): $P(\pi = [1\ 2\ 3]) = 0.4; P(\pi = [2\ 3\ 1]) = 0.3$; and $P(\pi = [3\ 1\ 2]) = 0.3$. The potential function $\psi_i(j) = 1$ if $i = j$ and 0 otherwise, provides a Bayes optimal permutation prediction for this distribution and an expected hinge loss of $3.6 = 0.4(3-3) + 0.3(3+3) + 0.3(3+3)$. However, the expected hinge loss is optimally minimized with a value of 3 when $\psi_i(j) = 0, \forall i, j$, which is indifferent between all permutations and is not Bayes optimal. Thus, Fisher consistency is not guaranteed.

3 Approach

To overcome the computational inefficiency of exponential random field methods and the Fisher inconsistency of maximum margin methods, we formulate the task of learning for bipartite matching problems as an adversarial structured prediction task. We present two approaches for efficiently solving the resulting game over permutations.

3.1 Permutation Mixture Formulation

The training data for bipartite matching consists of triplets (A, B, π) where A and B are two sets of nodes with equal size and π is the assignment. To simplify the notation, we denote x as the bipartite graph containing the nodes A and B. We also denote $\phi(x, \pi)$ as a vector that enumerates the joint feature representations based on the bipartite graph x and the matching assignment π. This joint feature is defined additively over each node assignment, i.e., $\phi(x, \pi) = \sum_{i=1}^n \phi_i(x, \pi_i)$.

Our approach seeks a predictor that robustly minimizes the Hamming loss against the worst-case permutation mixture probability that is consistent with the statistics of the training data. In this setting, a predictor makes a probabilistic prediction over the set of all possible assignments (denoted as $\hat{P}$). Instead of evaluating the predictor with the empirical distribution, the predictor is pitted against an adversary that also makes a probabilistic prediction (denoted as $\check{P}$). The predictor's objective is to minimize the expected loss function calculated from the predictor's and adversary's probabilistic predictions, while the adversary seeks to maximize the loss. The adversary (and only the adversary) is constrained to select a probabilistic prediction that matches the statistical summaries of the empirical training distribution (denoted as $\tilde{P}$) via moment matching constraints on joint features $\phi(x, \pi)$. Formally, we write our formulation as:

$$\min_{\hat{P}(\hat{\pi}|x)} \max_{\check{P}(\check{\pi}|x)} \mathbb{E}_{x\sim\tilde{P};\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi}, \check{\pi})\right] \text{ s.t.} \tag{2}$$
$$\mathbb{E}_{x\sim\tilde{P};\check{\pi}|x\sim\check{P}}\left[\sum_{i=1}^n \phi_i(x, \check{\pi}_i)\right] = \mathbb{E}_{(x,\pi)\sim\tilde{P}}\left[\sum_{i=1}^n \phi_i(x, \pi_i)\right].$$

This follows a recent line of work for adversarial classification under additive (Asif et al., 2015) and non-additive (Wang et al., 2015) loss functions that has been employed for chain-structured prediction (Li et al., 2016) and robust cut learning (Behpour et al., 2018). Using the method of Lagrangian multipliers and strong duality for convex-concave saddle point problems (Von Neumann & Morgenstern, 1945; Sion, 1958), The optimization in Eq. (2) can be equivalently solved in the dual formulation:

$$\min_{\theta} \mathbb{E}_{x,\pi\sim\tilde{P}} \min_{\hat{P}(\hat{\pi}|x)} \max_{\check{P}(\check{\pi}|x)} \mathbb{E}_{\substack{\hat{\pi}|x\sim\hat{P} \\ \check{\pi}|x\sim\check{P}}} \Bigg[\text{loss}(\hat{\pi}, \check{\pi}) + \tag{3}$$
$$\theta \cdot \sum_{i=1}^n \left(\phi_i(x, \check{\pi}_i) - \phi_i(x, \pi_i)\right)\Bigg],$$

where θ is the Lagrange dual variable for the moment matching constraints. We refer the reader to Appendix A in the supplementary materials for a more detailed explanation of this construction (i.e., the transformation from Eq. (2) to Eq. (3)). In this paper, we use Hamming distance, $\text{loss}(\hat{\pi}, \check{\pi}) = \sum_{i=1}^n 1(\hat{\pi}_i \neq \check{\pi}_i)$, as the loss function.

Table 1 shows the payoff matrix for the game of size $n = 3$ with 3! actions (permutations) for the predictor player $\hat{\pi}$ and for the adversarial approximation player $\check{\pi}$. Here, we define the difference between the Lagrangian potential of the adversary's action and the ground truth permutation as $\delta_{\check{\pi}} = \psi(\check{\pi}) - \psi(\pi) = \theta \cdot \sum_{i=1}^n \left(\phi_i(x, \check{\pi}_i) - \phi_i(x, \pi_i)\right)$.

Table 1. Augmented Hamming loss matrix for n=3 permutations.

	$\check{\pi} = 123$	$\check{\pi} = 132$	$\check{\pi} = 213$	$\check{\pi} = 231$	$\check{\pi} = 312$	$\check{\pi} = 321$
$\hat{\pi}=123$	$0 + \delta_{123}$	$2 + \delta_{132}$	$2 + \delta_{213}$	$3 + \delta_{231}$	$3 + \delta_{312}$	$2 + \delta_{321}$
$\hat{\pi}=132$	$2 + \delta_{123}$	$0 + \delta_{132}$	$3 + \delta_{213}$	$2 + \delta_{231}$	$2 + \delta_{312}$	$3 + \delta_{321}$
$\hat{\pi}=213$	$2 + \delta_{123}$	$3 + \delta_{132}$	$0 + \delta_{213}$	$2 + \delta_{231}$	$2 + \delta_{312}$	$3 + \delta_{321}$
$\hat{\pi}=231$	$3 + \delta_{123}$	$2 + \delta_{132}$	$2 + \delta_{213}$	$0 + \delta_{231}$	$3 + \delta_{312}$	$2 + \delta_{321}$
$\hat{\pi}=312$	$3 + \delta_{123}$	$2 + \delta_{132}$	$2 + \delta_{213}$	$3 + \delta_{231}$	$0 + \delta_{312}$	$2 + \delta_{321}$
$\hat{\pi}=321$	$2 + \delta_{123}$	$3 + \delta_{132}$	$3 + \delta_{213}$	$2 + \delta_{231}$	$2 + \delta_{312}$	$0 + \delta_{321}$

Unfortunately, the number of permutations, π, grows factorially ($\mathcal{O}(n!)$) with the number of elements being matched (n). This makes explicit construction of the Lagrangian minimax game intractable for modestly-sized problems.

3.2 Optimization by Constraint Generation

Our first approach for taming the factorial computational complexity of explicitly constructing games for larger matching tasks is a constraint-generation approach known as the *double oracle method* (McMahan et al., 2003). It obtains the equilibrium solution to the adversarial prediction game without explicitly constructing the entire game matrix (Table 1). Based on the key observation that the equilibrium of the zero-sum game is typically supported by a relatively small number of permutations, it seeks to efficiently uncover this sparse set of permutations for each player.

Algorithm 1 Double Oracle Algorithm for Adversarial Bipartite Matching Equilibria.

```
Input: Lagrangian potentials $\Psi(\cdot)$; Initial label $\pi_{\text{initial}}$
Output: The (sparse) Nash equilibrium $(\check{\mathcal{S}}, \hat{\mathcal{S}}, \hat{P}, \check{P})$
 1: $\check{\mathcal{S}} \leftarrow \hat{\mathcal{S}} \leftarrow \{\pi_{\text{initial}}\}$
 2: repeat
 3:     $(\hat{P}, \check{P}, \check{V}) \leftarrow \text{solveGame}(\boldsymbol{\Psi}(\check{\mathcal{S}}), \text{loss}_{\text{Ham}}(\hat{\mathcal{S}}, \check{\mathcal{S}}))$
 4:     $(\check{\pi}_{\text{new}}, V_{\text{max}}) \leftarrow \text{argmax}_{\check{\pi}} \mathbb{E}_{\hat{\pi} \sim \hat{P}}[\text{loss}_{\text{Ham}}(\hat{\pi}, \check{\pi}) + \Psi(\check{\pi})]$
 5:     if $(\check{V} \neq V_{\text{max}})$ then $\check{\mathcal{S}} \leftarrow \check{\mathcal{S}} \cup \check{\pi}_{\text{new}}$
 6:     $(\hat{P}, \check{P}, \hat{V}) \leftarrow \text{solveGame}(\boldsymbol{\Psi}(\check{\mathcal{S}}), \text{loss}_{\text{Ham}}(\hat{\mathcal{S}}, \check{\mathcal{S}}))$
 7:     $(\hat{\pi}_{\text{new}}, V_{\text{min}}) \leftarrow \text{argmin}_{\hat{\pi}} \mathbb{E}_{\check{\pi} \sim \check{P}}[\text{loss}_{\text{Ham}}(\hat{\pi}, \check{\pi})]$
 8:     if $(\hat{V} \neq V_{\text{min}})$ then $\hat{\mathcal{S}} \leftarrow \hat{\mathcal{S}} \cup \hat{\pi}_{\text{new}}$
 9: until $\check{V} = V_{\text{max}} = \hat{V} = V_{\text{min}}$
10: return $(\check{\mathcal{S}}, \hat{\mathcal{S}}, \hat{P}, \check{P})$
```

Algorithm 1 produces this set of "active" permutations for each player, $\hat{\mathcal{S}}$ and $\check{\mathcal{S}}$ (subsets of rows and columns in Table 1), and the associated Nash equilibrium $(\hat{P}, \check{P})$. Starting from an initial permutation, π_{initial} (Line 1), it repeatedly obtains the Nash equilibrium solution $(\hat{P}, \check{P})$ with value $\hat{V}$ or $\check{V}$ for the zero-sum game defined only by permutations in $\hat{\mathcal{S}}$ and $\check{\mathcal{S}}$ (Lines 3 and 6). This is efficiently accomplished using a linear program (Von Neumann & Morgenstern, 1945). The algorithm then obtains the other player's best response to either $\hat{P}$ or $\check{P}$ (Lines 4 and 7) with values V_{max} and V_{min} using the Kuhn-Munkres (Hungarian) algorithm in $\mathcal{O}(n^3)$ time for sets of size n. These best responses, $\check{\pi}_{\text{new}}$ and $\hat{\pi}_{\text{new}}$, are added to the set of active permutations (i.e., new rows or columns in the game matrix) if they have better values than the previous equilibrium values (Lines 5 and 8). This is repeated until no game value improvement exists for either player (Line 9), at which point a Nash equilibrium for the full game has been obtained.

We solve the convex optimization of Lagrange parameters θ in Eq. (3) using the results of Algorithm 1. We employ AdaGrad (Duchi et al., 2011) with the gradient calculated as the difference between expected features under the adversary's distribution and the empirical training data: $\mathbb{E}_{x \sim \tilde{P}; \check{\pi}|x \sim \check{P}} \left[\sum_{i=1}^{n} \phi_i(x, \check{\pi}_i)\right] - \mathbb{E}_{x, \pi \sim \tilde{P}} \left[\sum_{i=1}^{n} \phi_i(x, \pi_i)\right]$.

In contrast with SSVM, which compute the hinge loss for each training instance using only a single run of the Hungarian algorithm, our double oracle method must solve this problem repeatedly to find the equilibrium. Though in practice the total number of active permutations is much smaller than the $n!$ possibilities, no formal polynomial bound is known—and, consequentially, the run time of the approach as a whole cannot be characterized as polynomial.

3.3 Marginal Distribution Formulation

Our second approach, which significantly improves the efficiency of solving the adversarial bipartite matching game, leverages the key insight that all quantities of interest for evaluating the loss and satisfying the constraints depend only on marginal probabilities of the permutation's value assignments. Based on this, we employ a marginal distribution decomposition of the game.

Table 2. Doubly stochastic matrices $\mathbf{P}$ and $\mathbf{Q}$ for the marginal decompositions of each player's mixture of permutations.

	1	2	3
$\hat{\pi}_1$	$p_{1,1}$	$p_{1,2}$	$p_{1,3}$
$\hat{\pi}_2$	$p_{2,1}$	$p_{2,2}$	$p_{2,3}$
$\hat{\pi}_3$	$p_{3,1}$	$p_{3,2}$	$p_{3,3}$

	1	2	3
$\check{\pi}_1$	$q_{1,1}$	$q_{1,2}$	$q_{1,3}$
$\check{\pi}_2$	$q_{2,1}$	$q_{2,2}$	$q_{2,3}$
$\check{\pi}_3$	$q_{3,1}$	$q_{3,2}$	$q_{3,3}$

We begin this reformulation by first defining a matrix representation of permutation π as $\mathbf{Y}(\pi) \in \mathbb{R}^{n \times n}$ (or simply $\mathbf{Y}$) where the value of its cell $Y_{i,j}$ is 1 when $\pi_i = j$ and 0 otherwise. To be a valid complete bipartite matching or permutation, each column and row of $\mathbf{Y}$ can only have one entry of 1. For each feature function $\phi_i^{(k)}(x, \pi_i)$, we also denote its matrix representation as $\mathbf{X}_k$ whose (i, j)-th cell represents the k-th entry of $\phi_i(x, j)$. For a given distribution of permutations, $P(\pi)$, we denote the marginal probabilities of matching i with j as $p_{i,j} \triangleq P(\pi_i = j)$. We let $\mathbf{P} = \sum_{\pi} P(\pi) \mathbf{Y}(\pi)$ be the predictor's marginal probability matrix where its (i, j) cell represents $\hat{P}(\hat{\pi}_i = j)$, and similarly let $\mathbf{Q}$ be the adversary's marginal probability matrix (based on $\check{P}$), as shown in Table 2.

The Birkhoff–von Neumann theorem (Birkhoff, 1946; Von Neumann, 1953) states that the convex hull of the set of $n \times n$ permutation matrices forms a convex polytope in $\mathbb{R}^{n^2}$ (known as the Birkhoff polytope B_n) in which points are doubly stochastic matrices, i.e., the $n \times n$ matrices with non-negative elements where each row and column must sum to one. This implies that both marginal probability matrices $\mathbf{P}$ and $\mathbf{Q}$ are doubly stochastic matrices. In contrast to the space of distributions over permutation of n objects, which grows factorially ($\mathcal{O}(n!)$ with $n! - 1$ free parameters), the size of this marginal matrices grows only quadratically ($\mathcal{O}(n^2)$ with $n^2 - 2n$ free parameters). This provides a significant benefit in terms of the optimization.

Starting with the minimax over $\hat{P}(\hat{\pi})$ and $\check{P}(\check{\pi})$ in the permutation mixture formulation, and using the matrix notation above, we rewrite Eq. (3) as a minimax over marginal probability matrices $\mathbf{P}$ and $\mathbf{Q}$ with additional constraints that both $\mathbf{P}$ and $\mathbf{Q}$ are doubly-stochastic matrices, i.e., $\mathbf{P} \geq \mathbf{0}$ (elementwise), $\mathbf{Q} \geq \mathbf{0}$, $\mathbf{P1} = \mathbf{P}^\top \mathbf{1} = \mathbf{Q1} = \mathbf{Q}^\top \mathbf{1} = \mathbf{1}$ where $\mathbf{1} = (1, \dots, 1)^\top$). That is:

$$\min_{\theta} \mathbb{E}_{\mathbf{X}, \mathbf{Y} \sim \tilde{P}} \min_{\mathbf{P} \geq \mathbf{0}} \max_{\mathbf{Q} \geq \mathbf{0}} \left[n - \langle \mathbf{P}, \mathbf{Q} \rangle + \langle \mathbf{Q} - \mathbf{Y}, \textstyle\sum_k \theta_k \mathbf{X}_k \rangle\right]$$
$$\text{s.t.} : \mathbf{P1} = \mathbf{P}^\top \mathbf{1} = \mathbf{Q1} = \mathbf{Q}^\top \mathbf{1} = \mathbf{1}, \qquad (4)$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product between two matrices, i.e., $\langle A, B \rangle = \sum_{i,j} A_{i,j} B_{i,j}$.

3.3.1 Optimization

We reduce the computational costs of the optimization in Eq. (4) by focusing on optimizing the adversary's marginal probability $\mathbf{Q}$. By strong duality, we then push the maximization over $\mathbf{Q}$ in the formulation above to the outermost level of Eq. (4). Note that the objective above is a non-smooth function (i.e., piece-wise linear). For the purpose of smoothing the objective, we add a small amount of strongly convex prox-functions to both $\mathbf{P}$ and $\mathbf{Q}$. We also add a regularization penalty to the parameter θ to improve the generalizability of our model. We unfold Eq. (4) by replacing the empirical expectation with an average over all training examples, resulting in the following optimization:

$$\begin{aligned}\max_{\mathbf{Q}\geq\mathbf{0}} \min_{\theta} \frac{1}{m}\sum_{i=1}^{m} \min_{\mathbf{P}_i\geq\mathbf{0}} \Big[\langle \mathbf{Q}_i - \mathbf{Y}_i, \textstyle\sum_k \theta_k \mathbf{X}_{i,k}\rangle - \langle \mathbf{P}_i, \mathbf{Q}_i\rangle \\ + \tfrac{\mu}{2}\|\mathbf{P}_i\|_F^2 - \tfrac{\mu}{2}\|\mathbf{Q}_i\|_F^2\Big] + \tfrac{\lambda}{2}\|\theta\|_2^2 \\ \text{s.t.}: \mathbf{P}_i\mathbf{1} = \mathbf{P}_i^\top\mathbf{1} = \mathbf{Q}_i\mathbf{1} = \mathbf{Q}_i^\top\mathbf{1} = \mathbf{1}, \quad \forall i, \qquad (5)\end{aligned}$$

where m is the number of bipartite matching problems in the training set, λ is the regularization penalty parameter, μ is the smoothing penalty parameter, and $\|A\|_F$ denotes the Frobenius norm of matrix A. The subscript i in $\mathbf{P}_i, \mathbf{Q}_i, \mathbf{X}_i$, and $\mathbf{Y}_i$ refers to the i-th example in the training set.

In the formulation above, given a fixed $\mathbf{Q}$, the inner minimization over θ and $\mathbf{P}$ can then be solved separately. The optimal θ in the inner minimization admits a closed-form solution, in which the k-th element of θ^* is:

$$\theta_k^* = -\frac{1}{\lambda m}\sum_{i=1}^{m} \langle \mathbf{Q}_i - \mathbf{Y}_i, \mathbf{X}_{i,k}\rangle . \qquad (6)$$

The inner minimization over $\mathbf{P}$ can be solved independently for each training example. Given the adversary's marginal probability matrix $\mathbf{Q}_i$ for the i-th example, the optimal $\mathbf{P}_i$ can be formulated as:

$$\begin{aligned}\mathbf{P}_i^* &= \operatorname*{argmin}_{\{\mathbf{P}_i\geq\mathbf{0}|\mathbf{P}_i\mathbf{1}=\mathbf{P}_i^\top\mathbf{1}=\mathbf{1}\}} \tfrac{\mu}{2}\|\mathbf{P}_i\|_F^2 - \langle \mathbf{P}_i, \mathbf{Q}_i\rangle & (7)\\ &= \operatorname*{argmin}_{\{\mathbf{P}_i\geq\mathbf{0}|\mathbf{P}_i\mathbf{1}=\mathbf{P}_i^\top\mathbf{1}=\mathbf{1}\}} \|\mathbf{P}_i - \tfrac{1}{\mu}\mathbf{Q}_i\|_F^2. & (8)\end{aligned}$$

We can interpret this minimization as projecting the matrix $\frac{1}{\mu}\mathbf{Q}_i$ to the set of doubly-stochastic matrices. We will discuss our projection technique in the upcoming subsection.

For solving the outer optimization over $\mathbf{Q}$ with the doubly-stochastic constraints, we employ a projected Quasi-Newton algorithm (Schmidt et al., 2009). Each iteration of the algorithm optimizes the quadratic approximation of the objective function (using limited-memory Quasi-Newton) over the the convex set. In each update step, a projection to the set of doubly-stochastic matrices is needed, akin to the inner minimization of $\mathbf{P}$ in Eq. (8).

The optimization above provides the adversary's optimal marginal probability $\mathbf{Q}^*$. To achieve our learning goal, we recover θ^* using Eq. (6) computed over the optimal $\mathbf{Q}^*$. We use the θ^* that our model learns from this optimization to construct a weighted bipartite graph for making predictions for test examples.

3.3.2 Doubly-Stochastic Matrix Projection

The projection from an arbitrary matrix $\mathbf{R}$ to the set of doubly-stochastic matrices can be formulated as:

$$\min_{\mathbf{P}\geq\mathbf{0}} \|\mathbf{P} - \mathbf{R}\|_F^2, \qquad \text{s.t.}: \mathbf{P}\mathbf{1} = \mathbf{P}^\top\mathbf{1} = \mathbf{1}. \qquad (9)$$

We employ the alternating direction method of multipliers (ADMM) technique (Douglas & Rachford, 1956; Glowinski & Marroco, 1975; Boyd et al., 2011) to solve the optimization problem above. We divide the doubly-stochastic matrix constraint into two sets of constraints $C_1 : \mathbf{P}\mathbf{1} = \mathbf{1}$ and $\mathbf{P} \geq \mathbf{0}$, and $C_2 : \mathbf{P}^\top\mathbf{1} = \mathbf{1}$ and $\mathbf{P} \geq \mathbf{0}$. Using this construction, we convert the optimization above into ADMM form as follows:

$$\begin{aligned}&\min_{\mathbf{P},\mathbf{S}} \tfrac{1}{2}\|\mathbf{P} - \mathbf{R}\|_F^2 + \tfrac{1}{2}\|\mathbf{S} - \mathbf{R}\|_F^2 + \mathrm{I}_{C_1}(\mathbf{P}) + \mathrm{I}_{C_2}(\mathbf{S})\\ &\quad \text{s.t.}: \mathbf{P} - \mathbf{S} = 0. \qquad (10)\end{aligned}$$

The augmented Lagrangian for this optimization is:

$$\begin{aligned}\mathcal{L}_\rho(\mathbf{P},\mathbf{S},\mathbf{W}) =& \tfrac{1}{2}\|\mathbf{P} - \mathbf{R}\|_F^2 + \tfrac{1}{2}\|\mathbf{S} - \mathbf{R}\|_F^2 + \mathrm{I}_{C_1}(\mathbf{P})\\ &+ \mathrm{I}_{C_2}(\mathbf{S}) + \tfrac{\rho}{2}\|\mathbf{P} - \mathbf{S} + \mathbf{W}\|_F^2, \qquad (11)\end{aligned}$$

where ρ is the ADMM penalty parameter and $\mathbf{W}$ is the scaled dual variable. From the augmented Lagrangian, we compute the update for $\mathbf{P}$ as:

$$\begin{aligned}\mathbf{P}^{t+1} &= \operatorname*{argmin}_{\mathbf{P}} \mathcal{L}_\rho(\mathbf{P},\mathbf{S}^t,\mathbf{W}^t) \qquad (12)\\ &= \operatorname*{argmin}_{\{\mathbf{P}\geq\mathbf{0}|\mathbf{P}\mathbf{1}=\mathbf{1}\}} \tfrac{1}{2}\|\mathbf{P} - \mathbf{R}\|_F^2 + \tfrac{\rho}{2}\|\mathbf{P} - \mathbf{S}^t + \mathbf{W}^t\|_F^2\\ &= \operatorname*{argmin}_{\{\mathbf{P}\geq\mathbf{0}|\mathbf{P}\mathbf{1}=\mathbf{1}\}} \|\mathbf{P} - \tfrac{1}{1+\rho}\left(\mathbf{R} + \rho\left(\mathbf{S}^t - \mathbf{W}^t\right)\right)\|_F^2.\end{aligned}$$

The minimization above can be interpreted as a projection to the set $\{\mathbf{P} \geq 0|\mathbf{P}\mathbf{1} = \mathbf{1}\}$ which can be realized by projecting to the probability simplex independently for each row of the matrix $\frac{1}{1+\rho}\left(\mathbf{R} + \rho\left(\mathbf{S}^t - \mathbf{W}^t\right)\right)$. Similarly, the ADMM update for $\mathbf{S}$ can also be formulated as a column-wise probability simplex projection. The technique for projecting a point to the probability simplex has been studied previously, e.g., by Duchi et al. (2008). Therefore, our ADMM algorithm consists of the following updates:

$$\mathbf{P}^{t+1} = \mathrm{Proj}_{C_1}\left(\tfrac{1}{1+\rho}\left(\mathbf{R} + \rho\left(\mathbf{S}^t - \mathbf{W}^t\right)\right)\right) \qquad (13)$$

$$\mathbf{S}^{t+1} = \mathrm{Proj}_{C_2}\left(\tfrac{1}{1+\rho}\left(\mathbf{R} + \rho\left(\mathbf{P}^{t+1} + \mathbf{W}^t\right)\right)\right) \qquad (14)$$

$$\mathbf{W}^{t+1} = \mathbf{W}^t + \mathbf{P}^{t+1} - \mathbf{S}^{t+1}. \qquad (15)$$

We run this series of updates until the stopping conditions are met. Our stopping conditions are based on the primal and dual residual optimality as described in Boyd et al. (2011). In our overall algorithm, this ADMM projection algorithm is used both in the projected Quasi-Newton algorithm for optimizing $\mathbf{Q}$ (Eq. (5)) and in the inner optimization for minimizing $\mathbf{P}_i$ (Eq. (8)).

3.3.3 Convergence Property

The convergence rate of ADMM is $\mathcal{O}(\log \frac{1}{\epsilon})$ thanks to the strong convexity of the objective (Deng & Yin, 2016). Each step inside ADMM is simply a projection to a simplex, hence costing $\tilde{\mathcal{O}}(n)$ computations (Duchi et al., 2008).

In terms of optimization on $\mathbf{Q}$, since no explicit rates of convergence are available for the projected Quasi-Newton algorithm (Schmidt et al., 2009) that finely characterize the dependency on the condition numbers, we simply illustrate the $\sqrt{L/\mu}\log\frac{1}{\epsilon}$ rate using Nesterov's accelerated gradient algorithm (Nesterov, 2003), where L is the Lipschitz continuous constant of the gradient. In our case, $L = \frac{1}{m^2\lambda}\sum_k \sum_{i=1}^m \|\mathbf{X}_{i,k}\|_F^2 + 1/\mu$.

Comparison with Structured SVM (SSVM) Conventional SSVMs for learning bipartite matchings have only $\mathcal{O}(1/\epsilon)$ rates due to the lack of smoothness (Joachims et al., 2009; Teo et al., 2010). If smoothing is added, then similar linear convergence rates can be achieved with similar condition numbers. However, it is noteworthy that at each iteration we need to apply ADMM to solve a projection problem to the doubly stochastic matrix set (Eq. (9)), while SSVMs (without smoothing) solves a matching problem with the Hungarian algorithm, incurring $\mathcal{O}(n^3)$ time.

3.4 Consistency Analysis

Despite its apparent differences from standard empirical risk minimization (ERM), adversarial loss minimization (Eq. (3)) can be equivalently recast as an ERM:

$$\min_\theta \mathbb{E}_{\substack{x\sim P\\ \pi|x\sim\tilde{P}}} \left[AL^{\text{perm}}_{f_\theta}(x,\pi)\right] \text{ where } AL^{\text{perm}}_{f_\theta}(x,\pi) \triangleq$$

$$\min_{\hat{P}(\hat{\pi}|x)} \max_{\check{P}(\check{\pi}|x)} \mathbb{E}_{\substack{\hat{\pi}|x\sim\hat{P}\\ \check{\pi}|x\sim\check{P}}} \left[\text{loss}(\hat{\pi},\check{\pi}) + f_\theta(x,\check{\pi}) - f_\theta(x,\pi)\right]$$

and $f_\theta(x,\pi) = \theta \cdot \sum_{i=1}^n \phi(x,\pi_i)$ is the Lagrangian potential function. Here we consider f_θ as the linear discriminant function for a proposed permutation π, using parameter value θ. $AL^{\text{perm}}_{f_\theta}(x,\pi)$ is then the surrogate loss for input x and permutation π.

As described in Section 2.2, Fisher consistency is an important property for a surrogate loss L. It requires that under the true distribution $P(x,\pi)$, the hypothesis that minimizes L is Bayes optimal (Tewari & Bartlett, 2007; Liu, 2007). For the cases of multiclass classification and ordinal regression, Fisher consistency for adversarial surrogate loss has been established by Fathony et al. (2016; 2017). In our setting, the Fisher consistency of AL_f^{perm} can be written as

$$f^* \in \mathcal{F}^* \triangleq \operatorname*{argmin}_f \mathbb{E}_{\pi|x\sim P}\left[AL_f^{\text{perm}}(x,\pi)\right] \qquad (16)$$

$$\Rightarrow \operatorname*{argmax}_\pi f^*(x,\pi) \subseteq \Pi^\diamond \triangleq \operatorname*{argmin}_\pi \mathbb{E}_{\bar{\pi}|x\sim P}[\text{loss}(\pi,\bar{\pi})].$$

Note that in Eq. (16) we allow f to be optimized over the set of all measurable functions on the input space (x,π). In our formulation, we have restricted f to be additively decomposable over individual elements of permutation, $f(x,\pi) = \sum_i g_i(x,\pi_i)$. In the sequel, we will show that the condition in Eq. (16) also holds for this restricted set provided that g is allowed to be optimized over the set of all measurable functions on the space of individual input (x,π_i). We start by establishing Fisher consistency for the case of singleton loss minimizing sets $\Pi^\diamond$ in Theorem 1 and then for more general cases in Theorem 2.

Theorem 1. *Suppose loss*$(\pi,\bar{\pi})$ = *loss*$(\bar{\pi},\pi)$ *(symmetry) and loss*(π,π) < *loss*$(\bar{\pi},\pi)$ *for all* $\bar{\pi} \neq \pi$. *Then the adversarial permutation loss* AL_f^{perm} *is Fisher consistent if f is over all measurable functions and* $\Pi^\diamond$ *is a singleton.*

Theorem 2. *Suppose loss*$(\pi,\bar{\pi})$ = *loss*$(\bar{\pi},\pi)$ *(symmetry) and loss*(π,π) < *loss*$(\bar{\pi},\pi)$ *for all* $\bar{\pi} \neq \pi$. *Furthermore if f is over all measurable functions, then:*

(a) there exists $f^* \in \mathcal{F}^*$ *such that* $\operatorname{argmax}_\pi f^*(x,\pi) \subseteq \Pi^\diamond$ *(i.e., satisfies the Fisher consistency requirement). In fact, all elements in* $\Pi^\diamond$ *can be recovered by some* $f^* \in \mathcal{F}^*$.

(b) if $\operatorname{argmin}_\pi \sum_{\pi'\in\Pi^\diamond} \alpha_{\pi'}$*loss*$(\pi',\pi) \subseteq \Pi^\diamond$ *for all* $\alpha_{(\cdot)} \geq 0$; $\sum_{\pi'\in\Pi^\diamond} \alpha_{\pi'} = 1$, *then* $\operatorname{argmax}_\pi f^*(x,\pi) \subseteq \Pi^\diamond$ *for **all*** $f^* \in \mathcal{F}^*$. *In this case, all* $f^* \in \mathcal{F}^*$ *satisfy the Fisher consistency requirement.*

These assumptions of loss functions in the theorems above are quite mild, requiring only that wrong predictions suffer higher loss than correct ones. We refer the reader to Appendix B for the detailed proofs of theorems. The key to the proofs is the observation that for the optimal potential function f^*, $f^*(x,\pi) + \text{loss}(\pi,\pi^\diamond)$ is invariant to π when $\Pi^\diamond = \{\pi^\diamond\}$. We refer to this as the *loss reflective property*. Note that this generalizes the observation for the case of ordinal regression loss (Fathony et al., 2017) into matching loss functions, subject to the mild pre-conditions assumed by the theorem.

Theorem 3. *Suppose the loss is Hamming loss, and the potential function* $f(x,\pi)$ *decomposes additively by* $\sum_i g_i(x,\pi_i)$. *Then, the adversarial permutation loss* AL_f^{perm} *is Fisher consistent provided that* g_i *is allowed to*

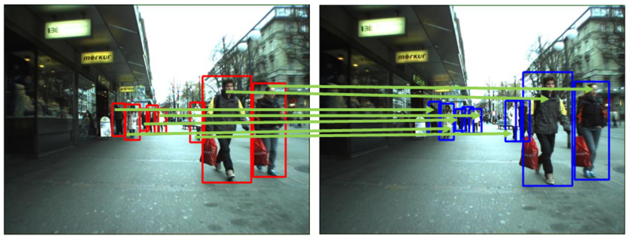

Figure 2. An example of bipartite matching in video tracking.

be optimized over the set of all measurable functions on the space of individual inputs (x, π_i).

Proof. Simply choose g_i such that for each sample x in the population, $g_i(x, \pi_i) = -(\pi_i \neq \pi_i^\diamond)$. This renders the loss reflective property under the Hamming loss. □

4 Experimental Evaluation

To evaluate our approach, we apply our adversarial bipartite matching model to video tracking tasks using public benchmark datasets (Leal-Taixé et al., 2015). In this problem, we are given a set of images (video frames) and a list of objects in each image. We are also given the correspondence matching between objects in frame t and objects in frame $t+1$. Figure 2 shows an example of the problem setup. It is important to note that the number of objects are not the same in every frames. Some of the objects may enter, leave, or remain in the consecutive frames. To handle the this issue, we setup our experiment as follows. Let k_t be the number of objects in frame t and k^* be the maximum number of objects a frame can have, i.e., $k^* = \max_{t \in T} k_t$. Starting from k^* nodes to represent the objects, we add k^* more nodes as "invisible" nodes to allow new objects to enter and existing objects to leave. As a result, the total number of nodes in each frame doubles to $n = 2k^*$.

4.1 Feature Representation

We define the features for pairs of bounding boxes (i.e., $\phi_i(x, j)$ for pairing bounding box i with bounding box j) in two consecutive video frames so that we can compute the associative feature vectors, $\phi(x, \pi) = \sum_{i=1}^{n} \phi_i(x, \pi_i)$, for each possible matching π. To define the feature vector $\phi_i(\cdot, \cdot)$, we follow the feature representation reported by Kim et al. (2012) using six different types of features:

- Intersection over union (IoU) overlap ratio between bounding boxes, $\text{area}(\text{BB}_i^t \cap \text{BB}_j^{t+1}) / \text{area}(\text{BB}_i^t \cup \text{BB}_j^{t+1})$, where BB_i^t denotes the bounding box of object i at time frame t;
- Euclidean distance between object centers;
- 21 color histogram distance features (RGB) from the Bhattacharyaa distance, $\frac{1}{4} \ln\left(\frac{1}{4}\left(\frac{\sigma_p^2}{\sigma_q^2} + \frac{\sigma_q^2}{\sigma_p^2} + 2\right)\right) + \frac{1}{4}\left(\frac{(\mu_p - \mu_q)^2}{\mu_p^2 + \mu_q^2}\right)$, between distributions from the histograms of 7×3 blocks, in which p and q are two different distributions of the blocks at time frames t and $t+1$, μ and σ^2 are the mean and the variance of the distribution respectively;
- 21 local binary pattern (LBP) features from similar Bhattacharyaa distances and bounding box blocks;
- Optical flow (motion) between bounding boxes; and
- Three indicator variables (for *entering*, *leaving*, and *staying invisible).*

Table 3. Dataset properties

DATASET	# ELEMENTS	# EXAMPLES
TUD-CAMPUS	12	70
TUD-STADTMITTE	16	178
ETH-SUNNYDAY	18	353
ETH-BAHNHOF	34	999
ETH-PEDCROSS2	30	836

We explain this feature representation in more detail in Appendix C.

4.2 Experimental Setup

We compare our approach with the Structured SVM (SSVM) model (Taskar et al., 2005a; Tsochantaridis et al., 2005) implemented based on Kim et al. (2012) using `SVM-Struct` (Joachims, 2008; Vedaldi, 2011). We implement our marginal version of adversarial bipartite matching using `minConf` (Schmidt, 2008) for performing projected Quasi-Newton optimization.

We consider two different groups of datasets in our experiment: TUD datasets and ETH datasets. Each dataset contains different numbers of elements (i.e., the number of pedestrian bounding box in the frame plus the number of extra nodes to indicate entering or leaving) and different numbers of examples (i.e., pairs of two consecutive frames that we want to match). Table 3 contains the detailed information about the datasets.

To avoid having test examples that are too similar with the training set, we train the models on one dataset and test the model on another dataset that has similar characteristics. In particular, we perform evaluations for every pair of datasets in TUD and ETH collections. This results in eight pairs of training/test datasets, as shown in Table 4.

To tune the regularization parameter (λ in adversarial matching, and C in SSVM), we perform 5-fold cross validation based on the training dataset only. The resulting best regularization parameter is used to train the model over all training examples to obtain parameters θ, which we then use to predict the matching for the testing data. For SSVM and the marginal version of adversarial matching, the pre-

Table 4. The mean and standard deviation (in parenthesis) of the average accuracy (1 - the average Hamming loss) for the adversarial bipartite matching model compared with Structured-SVM.

Training/ Testing	Adv DO	Adv Marg.	SSVM	Adv DO #Perm.
Campus/ Stadtmitte	0.662 (0.09)	0.662 (0.08)	0.662 (0.08)	11.4
Stadtmitte/ Campus	0.672 (0.12)	0.667 (0.11)	0.660 (0.12)	7.4
Bahnhof/ Sunnyday	**0.758** (0.12)	**0.754** (0.10)	0.729 (0.15)	5.8
Pedcross2/ Sunnyday	**0.760** (0.08)	**0.750** (0.10)	0.736 (0.13)	8.2
Sunnyday/ Bahnhof	**0.755** (0.20)	**0.751** (0.18)	0.739 (0.20)	9.8
Pedcross2/ Bahnhof	**0.760** (0.12)	**0.763** (0.16)	0.731 (0.21)	10.8
Bahnhof/ Pedcross2	**0.718** (0.16)	**0.714** (0.16)	0.701 (0.18)	8.5
Sunnyday/ Pedcross2	**0.719** (0.18)	**0.712** (0.17)	0.700 (0.18)	14.4

diction is done by finding the bipartite matching that maximizes the potential value, i.e., $\mathrm{argmax}_{\mathbf{Y}} \langle \mathbf{Y}, \sum_k \theta_k \mathbf{X}_k \rangle$ which can be solved using the Hungarian algorithm. The double oracle version of adversarial matching makes predictions by finding the most likely permutation from the predictor's strategy in the equilibrium.

4.3 Results

We report the average accuracy, which in this case is defined as ($1 -$ the average Hamming loss) over all examples in the testing dataset. Table 4 shows the mean and the standard deviation of our metric across different dataset pairs. We report the results for both the double-oracle (DO) and marginal (Marg) versions of the adversarial model. Our experiment indicates that both methods result in very similar values of θ. The slight advantage of the double-oracle version is caused by the difference in prediction techniques between the double-oracle (argmax over predictor's equilibrium strategy) and marginal version (argmax over potentials). We also observe that the double-oracle approach requires only a small number of augmenting permutations to converge as shown in the last column (the average number of permutations) of Table 4. This indicates the sparseness of the set of permutations that support the equilibrium.

To compare with SSVM, we highlight (using bold font) the cases in which our result is better with statistical significance (under paired t-test with $\alpha < 0.05$) in Table 4. Compared with SSVM, our proposed adversarial matching outperforms SSVM in all pairs of datasets—with statistical significance on all six pairs of the ETH datasets and slightly better than SSVM on the TUD datasets. This suggests that our adversarial bipartite matching model benefits from its Fisher consistency property.

Table 5. Running time (in seconds) of the model for various number of elements n with fixed number of samples ($m = 50$)

Dataset	# Elements	Adv Marg.	SSVM
Campus	12	1.96	0.22
Stadtmitte	16	2.46	0.25
Sunnyday	18	2.75	0.15
Pedcross2	30	8.18	0.26
Bahnhof	34	9.79	0.31

In terms of the running time, Table 5 shows that the marginal version of adversarial method is relatively fast. It only takes a few seconds to train until convergence in the case of 50 examples, with the number of elements varied up to 34. The running time grows roughly quadratically in the number of elements, which is natural since the size of the marginal probability matrices $\mathbf{P}$ and $\mathbf{Q}$ also grow quadratically in the number of elements. This shows that our approach is much more efficient than the CRF approach, which has a running time that is impractical even for small problems with 20 elements. The training time of SSVM is faster than the adversarial methods due to two different factors: (1) the inner optimization of SSVM can be solved using a single execution of the Hungarian algorithm compared with the inner optimization of adversarial method which requires ADMM optimization for projection to doubly stochastic matrix set; (2) different tools for implementation, i.e., C++ for SSVM and MATLAB for our method, which benefits the running time of SSVM. In addition, though the game size is relatively small, as indicated by the final column in Table 4, the double oracle version of adversarial method takes much longer to train compared to the marginal version.

5 Conclusions and Future Work

In this paper, we have presented an adversarial approach for learning bipartite matchings that is not only computationally efficient to employ but also provides Fisher consistency guarantees. We showed that these theoretical advantages translate into better empirical performance for our model compared with previous approaches. Our future work will explore matching problems with different loss functions and other graphical structures.

Acknowledgements

We thank our anonymous reviewers for their useful feedback and suggestions. This research was supported in part by NSF Grants RI-#1526379 and CAREER-#1652530.

References

Ahonen, T., Hadid, A., and Pietikainen, M. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

Amini, M. R., Truong, T. V., and Goutte, C. A boosting algorithm for learning bipartite ranking functions with partially labeled data. In *Proceedings of the International ACM SIGIR Conference*, pp. 99–106. ACM, 2008.

Asif, K., Xing, W., Behpour, S., and Ziebart, B. D. Adversarial cost-sensitive classification. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 92–101, 2015.

Beauchemin, S. S. and Barron, J. L. The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3): 433–466, 1995.

Behpour, S., Xing, W., and Ziebart, B. D. ARC: Adversarial robust cuts for semi-supervised and multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2704–2711, 2018.

Belongie, S., Malik, J., and Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (4):509–522, 2002.

Birkhoff, G. Three observations on linear algebra. *Univ. Nac. Tacuman, Rev. Ser. A*, 5:147–151, 1946.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Chan, Y. S. and Ng, H. T. Maxsim: A maximum similarity metric for machine translation evaluation. *Proceedings of ACL-08: HLT*, pp. 55–62, 2008.

Dellaert, F., Seitz, S. M., Thorpe, C. E., and Thrun, S. EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine Learning*, 50 (1-2):45–71, 2003.

Deng, W. and Yin, W. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66(3), 2016.

Douglas, J. and Rachford, H. H. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439, 1956.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the International Conference on Machine Learning*, pp. 272–279. ACM, 2008.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. Rank aggregation methods for the web. In *Proceedings of the International Conference on World Wide Web*, pp. 613–622. ACM, 2001.

Fathony, R., Liu, A., Asif, K., and Ziebart, B. Adversarial multiclass classification: A risk minimization perspective. In *Advances in Neural Information Processing Systems*, pp. 559–567, 2016.

Fathony, R., Bashiri, M. A., and Ziebart, B. Adversarial surrogate losses for ordinal regression. In *Advances in Neural Information Processing Systems*, pp. 563–573, 2017.

Glowinski, R. and Marroco, A. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2):41–76, 1975.

Grünwald, P. D. and Dawid, A. P. Game theory, maximum entropy, minimum discrepancy, and robust Bayesian decision theory. *Annals of Statistics*, 32:1367–1433, 2004.

Huber, M. and Law, J. Fast approximation of the permanent for very dense problems. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 681–689. Society for Industrial and Applied Mathematics, 2008.

Joachims, T. SVM-struct: Support vector machine for complex outputs. `http://www.cs.cornell.edu/People/tj/svm_light/svm_struct.html`, 2008.

Joachims, T., Finley, T., and Yu, C.-N. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, 2009.

Kim, S., Kwak, S., Feyereisl, J., and Han, B. Online multi-target tracking by large margin structured learning. In *Asian Conference on Computer Vision*, pp. 98–111. Springer, 2012.

Kuhn, H. W. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97, 1955.

Lafferty, J., McCallum, A., and Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the International Conference on Machine Learning*, pp. 282–289, 2001.

Le, Q. and Smola, A. Direct optimization of ranking measures. *arXiv preprint arXiv:0704.3359*, 2007.

Leal-Taixé, L., Milan, A., Reid, I., Roth, S., and Schindler, K. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015.

Li, J., Asif, K., Wang, H., Ziebart, B. D., and Berger-Wolf, T. Y. Adversarial sequence tagging. In *International Joint Conference on Artificial Intelligence*, 2016.

Liu, L., Sun, L., Rui, Y., Shi, Y., and Yang, S. Web video topic discovery and tracking via bipartite graph reinforcement model. In *Proceedings of the 17th International Conference on World Wide Web*, pp. 1009–1018. ACM, 2008.

Liu, Y. Fisher consistency of multicategory support vector machines. In *International Conference on Artificial Intelligence and Statistics*, pp. 291–298, 2007.

MacCartney, B., Galley, M., and Manning, C. D. A phrase-based alignment model for natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 802–811. Association for Computational Linguistics, 2008.

McMahan, H. B., Gordon, G. J., and Blum, A. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the International Conference on Machine Learning*, pp. 536–543, 2003.

Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2003.

Padó, S. and Lapata, M. Optimal constituent alignment with edge covers for semantic projection. In *Proceedings of the International Conference on Computational Linguistics*, pp. 1161–1168. Association for Computational Linguistics, 2006.

Petterson, J., Yu, J., McAuley, J. J., and Caetano, T. S. Exponential family graph matching and ranking. In *Advances in Neural Information Processing Systems*, pp. 1455–1463, 2009.

Rui, X., Li, M., Li, Z., Ma, W.-Y., and Yu, N. Bipartite graph reinforcement model for web image annotation. In *Proceedings of the 15th ACM International Conference on Multimedia*, pp. 585–594. ACM, 2007.

Schmidt, M. minConf: projection methods for optimization with simple constraints in Matlab. `http://www.cs.ubc.ca/~schmidtm/Software/minConf.html`, 2008.

Schmidt, M., Berg, E., Friedlander, M., and Murphy, K. Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm. In *Artificial Intelligence and Statistics*, pp. 456–463, 2009.

Sion, M. On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176, 1958.

Taskar, B., Chatalbashev, V., Koller, D., and Guestrin, C. Learning structured prediction models: A large margin approach. In *Proceedings of the International Conference on Machine Learning*, pp. 896–903. ACM, 2005a.

Taskar, B., Lacoste-Julien, S., and Klein, D. A discriminative matching approach to word alignment. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 73–80. Association for Computational Linguistics, 2005b.

Taylor, W. R. Protein structure comparison using bipartite graph matching and its application to protein structure classification. *Molecular & Cellular Proteomics*, 1(4): 334–339, 2002.

Teo, C. H., Vishwanthan, S. V. N., Smola, A. J., and Le, Q. V. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11:311–365, January 2010.

Tewari, A. and Bartlett, P. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

Topsøe, F. Information-theoretical optimization techniques. *Kybernetika*, 15(1):8–27, 1979.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(Sep):1453–1484, 2005.

Valiant, L. G. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

Vedaldi, A. A MATLAB wrapper of SVM$^{\text{struct}}$. `http://www.vlfeat.org/~vedaldi/code/svm-struct-matlab.This`, 2011.

Volkovs, M. and Zemel, R. S. Efficient sampling for bipartite matching problems. In *Advances in Neural Information Processing Systems*, pp. 1313–1321, 2012.

Von Neumann, J. A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games*, 2:5–12, 1953.

Von Neumann, J. and Morgenstern, O. Theory of games and economic behavior. *Bull. Amer. Math. Soc*, 51(7): 498–504, 1945.

Wang, H., Xing, W., Asif, K., and Ziebart, B. Adversarial prediction games for multivariate losses. In *Advances in Neural Information Processing Systems*, pp. 2710–2718, 2015.

Wang, X.-Y., Wu, J.-F., and Yang, H.-Y. Robust image retrieval based on color histogram of local feature regions. *Multimedia Tools and Applications*, 49(2):323–345, 2010.

Wang, Y., Makedon, F., Ford, J., and Huang, H. A bipartite graph matching framework for finding correspondences between structural elements in two proteins. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pp. 2972–2975, Sept 2004.

Zhu, J., Hoi, S. C., Lyu, M. R., and Yan, S. Near-duplicate keyframe retrieval by nonrigid image matching. In *Proceedings of the 16th ACM international conference on Multimedia*, pp. 41–50. ACM, 2008.

Appendix A. Dual Permutation Mixture Formulation

Below is the detailed step-by-step transformation from the primal mixture formulation of the adversarial prediction task for bipartite matching (Eq. (2)) to the dual formulation (Eq. (3)):

$$\min_{\hat{P}(\hat{\pi}|x)} \max_{\check{P}(\check{\pi}|x)} \mathbb{E}_{x\sim\tilde{P};\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi})\right] \quad \text{s.t.} \quad \mathbb{E}_{x\sim\tilde{P};\check{\pi}|x\sim\check{P}} \left[\sum_{i=1}^{n} \phi_i(x,\check{\pi}_i)\right] = \mathbb{E}_{(x,\pi)\sim\tilde{P}} \left[\sum_{i=1}^{n} \phi_i(x,\pi_i)\right] \tag{17}$$

$$\overset{(a)}{=} \max_{\check{P}(\check{\pi}|x)} \min_{\hat{P}(\hat{\pi}|x)} \mathbb{E}_{x\sim\tilde{P};\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi})\right] \quad \text{s.t.} \quad \mathbb{E}_{x\sim\tilde{P};\check{\pi}|x\sim\check{P}} \left[\sum_{i=1}^{n} \phi_i(x,\check{\pi}_i)\right] = \mathbb{E}_{(x,\pi)\sim\tilde{P}} \left[\sum_{i=1}^{n} \phi_i(x,\pi_i)\right] \tag{18}$$

$$\overset{(b)}{=} \max_{\check{P}(\check{\pi}|x)} \min_{\theta} \min_{\hat{P}(\hat{\pi}|x)} \mathbb{E}_{(x,\pi)\sim\tilde{P};\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi}) + \theta^{\mathrm{T}} \left(\sum_{i=1}^{n} \phi_i(x,\check{\pi}_i) - \sum_{i=1}^{n} \phi_i(x,\pi_i)\right)\right] \tag{19}$$

$$\overset{(c)}{=} \min_{\theta} \max_{\check{P}(\check{\pi}|x)} \min_{\hat{P}(\hat{\pi}|x)} \mathbb{E}_{(x,\pi)\sim\tilde{P};\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi}) + \theta^{\mathrm{T}} \left(\sum_{i=1}^{n} \phi_i(x,\check{\pi}_i) - \sum_{i=1}^{n} \phi_i(x,\pi_i)\right)\right] \tag{20}$$

$$\overset{(d)}{=} \min_{\theta} \mathbb{E}_{(x,\pi)\sim\tilde{P}} \max_{\check{P}(\check{\pi}|x)} \min_{\hat{P}(\hat{\pi}|x)} \mathbb{E}_{\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi}) + \theta \cdot \sum_{i=1}^{n} \left(\phi_i(x,\check{\pi}_i) - \phi_i(x,\pi_i)\right)\right] \tag{21}$$

$$\overset{(e)}{=} \min_{\theta} \mathbb{E}_{(x,\pi)\sim\tilde{P}} \min_{\hat{P}(\hat{\pi}|x)} \max_{\check{P}(\check{\pi}|x)} \mathbb{E}_{\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi}) + \theta \cdot \sum_{i=1}^{n} \left(\phi_i(x,\check{\pi}_i) - \phi_i(x,\pi_i)\right)\right]. \tag{22}$$

The transformation steps above are described in the following:

(a) Flipping the min and max order using the minimax duality (Von Neumann & Morgenstern, 1945).

(b) Introducing the Lagrange dual variable θ.

(c) The domain of $\check{P}(\check{\pi}|x)$ is a compact convex set (i.e., permutation mixture distribution), whereas the domain of θ is convex (i.e., $\mathbb{R}^d$ where d is the number of features). The objective is concave on $\check{P}(\check{\pi}|x)$ since a non-negative linear combination of minimums of affine function is concave, while it is convex on θ. Sion's minimax theorem (Sion, 1958) says that a strong duality holds. Therefore, we can flip the order of $\check{P}(\check{\pi}|x)$ and θ in the optimization.

(d) Pushing the expectation over the empirical distribution outside the inner maximin, and changing the vector multiplication notation into a vector dot product.

(e) Applying the minimax duality (Von Neumann & Morgenstern, 1945) again to flip the optimization order of the inner minimax, resulting in Eq. (3).

Appendix B. Proofs for the Consistency Analysis

B.1 Proof of Theorem 1

Theorem 1. *Suppose loss$(\pi,\bar{\pi})$ = loss$(\bar{\pi},\pi)$ (symmetry) and loss$(\pi,\pi) <$ loss$(\bar{\pi},\pi)$ for all $\bar{\pi} \neq \pi$. Then the adversarial permutation loss AL_f^{perm} is Fisher consistent if f is over all measurable functions and $\Pi^\diamond$ is a singleton.*

Proof. Denote $\mathbf{p}$ as the probability mass given by the predictor player $\hat{P}(\hat{\pi}|x)$, $\mathbf{q}$ as the probability mass given by the adversary player $\check{P}(\check{\pi}|x)$, and $\mathbf{d}$ as the probability mass of the true distribution $P(\pi|x)$. So, all $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{d}$ lie in the $n!$ dimensional probability simplex Δ. Let C be an $n!$-by-$n!$ matrix whose $(\pi,\bar{\pi})$-th entry is loss$(\pi,\bar{\pi})$. Let $\mathbf{f} \in \mathbb{R}^{n!}$ the vector encoding of the value of f at all permutations. The definition of f^* in Eq. (16) now becomes:

$$\mathbf{f}^* \in \operatorname*{argmin}_{\mathbf{f}} \max_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \left\{\mathbf{f}^\top\mathbf{q} + \mathbf{p}^\top C\mathbf{q} - \mathbf{d}^\top\mathbf{f}\right\} \tag{23}$$

$$= \operatorname*{argmin}_{\mathbf{f}} \max_{\mathbf{q}\in\Delta} \left\{\mathbf{f}^\top\mathbf{q} + \min_{\pi}(C\mathbf{q})_\pi - \mathbf{d}^\top\mathbf{f}\right\}. \tag{24}$$

Let $\Pi^\diamond = \operatorname{argmin}_\pi \mathbb{E}_{\bar\pi|x\sim P}[\text{loss}(\pi,\bar\pi)]$ (or equivalently $\operatorname{argmin}_\pi (C\mathbf{d})_\pi$) contains only a singleton which we denote as $\pi^\diamond$. We are to show that $\operatorname{argmax}_\pi f^*(x,\pi)$ is a singleton, and its only element π^* is exactly $\pi^\diamond$. Since $\mathbf{f}^*$ is an optimal solution, the objective function must have a zero subgradient at $\mathbf{f}^*$. That means $\mathbf{0} = \mathbf{q}^* - \mathbf{d}$, where $\mathbf{q}^*$ is an optimal solution in Eq. (24) under $\mathbf{f}^*$. As a result:

$$\mathbf{d} \in \operatorname*{argmax}_{\mathbf{q}\in\Delta} \left\{ \mathbf{q}^\top \mathbf{f}^* + \min_\pi (C\mathbf{q})_\pi \right\}. \tag{25}$$

By the first order optimality condition of constrained convex optimization (see Eq. (4.21) of (Boyd & Vandenberghe, 2004)), this means (let $C_{:,\pi^\diamond}$ be the $\pi^\diamond$-th column of C):

$$(\mathbf{f}^* + C_{:,\pi^\diamond})^\top (\mathbf{u} - \mathbf{d}) \leq 0 \quad \forall \mathbf{u} \in \Delta, \tag{26}$$

where $\mathbf{f}^* + C_{:,\pi^\diamond}$ is the gradient of the objective in Eq. (25) with respect to $\mathbf{q}$ evaluated at $\mathbf{q} = \mathbf{d}$. Here we used the definition of $\pi^\diamond$. However, this inequality can hold for some $\mathbf{d} \in \Delta \cap \mathbb{R}^{n!}_{++}$ only if $\mathbf{f}^* + C_{:,\pi^\diamond}$ is a uniform vector, i.e., $f^*_\pi + \text{loss}(\pi, \pi^\diamond)$ is a constant in π. To see this, let's assume the contrary that $\mathbf{v} \triangleq \mathbf{f}^* + C_{:,\pi^\diamond}$ is not a uniform vector, and let j be the index of its maximum element. Let $\mathbf{u}$ be a vector whose values are 1 for index j and 0 otherwise. It is clear that for any $\mathbf{d} \in \Delta \cap \mathbb{R}^{n!}_{++}$, $\mathbf{v}^\top\mathbf{u} > \mathbf{v}^\top\mathbf{d}$, and hence $(\mathbf{f}^* + C_{:,\pi^\diamond})^\top(\mathbf{u} - \mathbf{d}) > 0$.

Finally, using the assumption that $\text{loss}(\pi,\pi) < \text{loss}(\bar\pi,\pi)$ for all $\bar\pi \neq \pi$, it follows that $\pi^* = \pi^\diamond$, since $\operatorname{argmax}_\pi f^*(x,\pi) = \operatorname{argmin}_\pi (C_{:,\pi^\diamond})_\pi$. □

B.2 Proof of Theorem 2

Theorem 2. *Suppose loss$(\pi,\bar\pi)$ = loss$(\bar\pi,\pi)$ (symmetry) and loss(π,π) < loss$(\bar\pi,\pi)$ for all $\bar\pi \neq \pi$. Furthermore if f is over all measurable functions, then:*

(a) there exists $f^* \in \mathcal{F}^*$ such that $\operatorname{argmax}_\pi f^*(x,\pi) \subseteq \Pi^\diamond$ (i.e., satisfies the Fisher consistency requirement). In fact, all elements in $\Pi^\diamond$ can be recovered by some $f^* \in \mathcal{F}^*$.

*(b) if $\operatorname{argmin}_\pi \sum_{\pi'\in\Pi^\diamond} \alpha_{\pi'}$loss$(\pi',\pi) \subseteq \Pi^\diamond$ for all $\alpha_{(\cdot)} \geq 0$; $\sum_{\pi'\in\Pi^\diamond} \alpha_{\pi'} = 1$, then $\operatorname{argmax}_\pi f^*(x,\pi) \subseteq \Pi^\diamond$ for **all** $f^* \in \mathcal{F}^*$. In this case, all $f^* \in \mathcal{F}^*$ satisfy the Fisher consistency requirement.*

Proof. Let $\Pi^\diamond$ be the set containing all of the solution of $\operatorname{argmin}_\pi (C\mathbf{d})_\pi$, i.e., $\Pi^\diamond = \{\pi^\diamond \mid (C\mathbf{d})_{\pi^\diamond} = \min_\pi (C\mathbf{d})_\pi\}$. The analyses in the proof of Theorem 1 still apply to this case, except for the Eq. (26). Denote $h(\mathbf{q}) \triangleq \mathbf{q}^\top\mathbf{f}^* + \min_\pi (C\mathbf{q})_\pi$. The sub-differential of $h(\mathbf{q})$ evaluated at $\mathbf{q} = \mathbf{d}$ is the set:

$$\partial h(\mathbf{d}) = \{\mathbf{f}^* + \mathbf{v} \mid \mathbf{v} \in \mathbf{conv}\{C_{:,\pi^\diamond} \mid \pi^\diamond \in \Pi^\diamond\}\}, \tag{27}$$

where **conv** denotes the convex hull of a finite point set. By extending the first order optimality condition to the subgradient case, this means that there is a subgradient $\mathbf{g} \in \partial h(\mathbf{d})$ such that:

$$\mathbf{g}^\top (\mathbf{u} - \mathbf{d}) \leq 0 \quad \forall \mathbf{u} \in \Delta. \tag{28}$$

Similar to the singleton $\Pi^\diamond$ case, this inequality can hold for some $\mathbf{d} \in \Delta \cap \mathbb{R}^{n!}_{++}$ only if $\mathbf{g}$ is a uniform vector. Based on Eq. (27), $\mathbf{g} - \mathbf{f}^*$ can be written as a convex combination of the elements in $\Pi^\diamond$, and thus:

$$\mathbf{f}^* = k\mathbf{1} - \sum_{\pi'\in\Pi^\diamond} \alpha_{\pi'} C_{:,\pi'}, \tag{29}$$

for some set of $\alpha_{(\cdot)} \geq 0$, $\sum_{\pi'\in\Pi^\diamond} \alpha_{\pi'} = 1$ and some constant k. This means that multiple solutions of $\mathbf{f}^*$ are possible. Let us denote the set of containing all solutions as $\mathcal{F}^*$. For each element $\pi^\diamond$ in $\Pi^\diamond$, we can recover a $f^*_{\pi^\diamond}$ in which the $\operatorname{argmax}_\pi f^*_{\pi^\diamond}(x,\pi)$ contains a singleton element $\pi^\diamond$ by using Eq. (29) with $\alpha_{\pi^\diamond} = 1$ and $\alpha_{\pi'\in\{\Pi^\diamond\setminus\pi^\diamond\}} = 0$. This is implied by our loss assumption that $\text{loss}(\pi,\pi) < \text{loss}(\bar\pi,\pi)$ for all $\bar\pi \neq \pi$, and hence $\operatorname{argmax}_\pi f^*_{\pi^\diamond}(x,\pi) = \operatorname{argmin}_\pi (C_{:,\pi^\diamond})_\pi$.

Furthermore, if we add another assumption on the loss function such that $\operatorname{argmin}_\pi \sum_{\pi'\in\Pi^\diamond} \alpha_{\pi'} \text{loss}(\pi',\pi) \subseteq \Pi^\diamond$ for all $\alpha_{(\cdot)} \geq 0$, $\sum_{\pi'\in\Pi^\diamond} \alpha_{\pi'} = 1$, then it follows that $\operatorname{argmax}_\pi f^*(x,\pi) \subseteq \Pi^\diamond$ for all $f^* \in \mathcal{F}^*$, since for any loss function that satisfy the assumption, $\operatorname{argmin}_\pi \left(\sum_{\pi'\in\Pi^\diamond} a_{\pi'} C_{:,\pi'}\right)_\pi \subseteq \Pi^\diamond$ for all $\alpha_{(\cdot)} \geq 0$, $\sum_{\pi'\in\Pi^\diamond} \alpha_{\pi'} = 1$. □

Appendix C. Feature Representation

We explain local binary pattern (LBP), color histogram (RGB), and optical flow within our feature representation in more details in this section. LBP is one of the best and most widely used texture descriptors in different applications like face detection. It assigns a label to every pixel of an image by thresholding the 3×3 neighborhood of each pixel with the center pixel value and reporting a binary number as the result. It is discriminative and invariant to monotonic gray-level changes (Ahonen et al., 2006). An important element in content-based image retrieval is the image color. Global histogram is one of the most popular color information representations. It presents the joint distribution of intensities of three-color (Red, Green, and Blue) channels. Its robustness to background complications and object distortion provides helpful hints for the subsequent expression of similarity between images (Wang et al., 2010).

To extract LBP and color of histograms features, we first divide the object regions to 7×3 blocks based on the aspect ratio of the detected pedestrians in the dataset, which is also 7:3. For each block, we calculate the distribution of LBP and then employ the Bhattacharyya coefficient to compute the affinity of a pair of distributions. Bhattacharyya coefficient (BC) measures the amount of overlap between two distributions. It returns 21 features for LBP. For color histogram, we represent the color information of each block using a 3D RGB color histogram of $8\times8\times8$ dimension. Then BC is applied and 21 RGB features are extracted.

Optical flow is an image motion representation and defined as the projection of velocities of 2D/3D surface points. It is based on correspondences between image features, correlations, or properties of intensity structures (Beauchemin & Barron, 1995). We compute the histogram of optical flow (HOF) for every detected box and employ BC to calculate the motion distribution relation. It returns one feature as optical flow.

We also consider three binary features (entering, leaving, and staying invisible) to indicate the status of each object between two consecutive frames.

Together, each feature vector, $\phi_i(x, j)$, has 48 values.